Deep Learning Solution for Dynamic Corporate Finance Models

Yizhou Li

November 24, 2025

Abstract

This report presents a deep learning-based solution to the dynamic investment model described in Strebulaev (2012). We address the non-convexities introduced by fixed adjustment costs using the method proposed by Maliar et al. (2021). By utilizing the "All-in-One" (AiO) expectation operator and a novel curriculum learning strategy for discontinuity smoothing, we successfully capture the lumpy investment behavior predicted by economic theory.

1 Literature Review

Dynamic investment under uncertainty is a central theme in corporate finance. As surveyed by Strebulaev and Whited (2012), financial frictions—particularly fixed adjustment costs—fundamentally alter firm behavior. These frictions create a non-convex optimization problem characterized by an "inaction region" where firms refrain from investing until productivity shocks cross a specific threshold. Traditional static models fail to capture these dynamics, and standard linearized DSGE methods are ill-suited for such discrete choices.

Solving these models numerically is computationally demanding. Grid-based methods, such as Value Function Iteration (VFI), suffer from the "curse of dimensionality" as the state space expands. Recently, Maliar, Maliar, and Winant (2021) introduced a deep learning framework that recasts the system of Bellman and Euler equations into a stochastic loss minimization problem. Their "All-in-One" (AiO) expectation operator allows for efficient handling of high-dimensional integrals via stochastic gradient descent (SGD), providing a scalable alternative to traditional projection methods.

2 Choice of Methods and Justification

We adopt the deep learning-based Euler Equation method with the AiO operator (Maliar et al., 2021) as our primary solver.

Reason for Choice:

- **Scalability:** Unlike grid-based methods where computational cost grows exponentially with the number of state variables, this method scales linearly. This is critical for extending the framework to more sophisticated settings, such as the Risky Debt model (Strebulaev, 2012, Section 3.6), which introduces additional state variables (e.g., debt levels, default boundaries).

- **Handling Discontinuities:** The fixed adjustment cost in our target model introduces a discontinuity at zero investment ($I = 0$). Standard derivative-based solvers struggle with this non-convexity. The deep learning approach allows us to implement a smooth approximation strategy (detailed in Section 3), enabling the use of gradient descent to find the optimal policy effectively.

3 Solution and Analysis

3.1 Model Implementation

We focus on the basic dynamic investment model (Strebulaev, Section 3.1) where a firm maximizes shareholder value subject to adjustment costs. The firm's production function is Cobb-Douglas with decreasing returns to scale:

$$\pi(k, z) = zk^{\theta} \tag{1}$$

where k is capital, z is the stochastic productivity shock following an AR(1) process, and θ is the curvature parameter. The capital accumulation follows $k' = (1 - \delta)k + I$.

The core challenge lies in the adjustment cost function $\psi(I, k)$, which includes both convex and fixed components:

$$\psi(I, k) = \frac{\psi_0}{2}\frac{I^2}{k} + \psi_1 k \cdot \mathbb{I}(I \neq 0) \tag{2}$$

where $\mathbb{I}(\cdot)$ is the indicator function.

We implemented the solution using TensorFlow 2.x with two deep neural networks: a Value Network approximating $V(k, z)$ and a Policy Network approximating the optimal capital rule $k' = \phi(k, z)$. The objective is to minimize the weighted sum of squared Bellman residuals (R_B) and Euler residuals (R_E):

$$\mathcal{L} = \mathbb{E}_{\omega}\left[R^2_{Bellman} + \nu R^2_{Euler}\right] \tag{3}$$

We utilized the AiO operator with two uncorrelated shocks (z'_1, z'_2) to ensure unbiased gradient estimation for the squared terms.

3.2 Synthetic Data and Effectiveness

To test the solution, we generated a set of synthetic data by simulating the economy for 100 periods using the trained policy network. We define "effectiveness" based on two criteria:

1. **Economic Consistency:** The model must reproduce stylized facts, specifically "lumpy" investment.

2. **Numerical Accuracy:** The solution must satisfy the theoretical first-order conditions (Euler Equation).

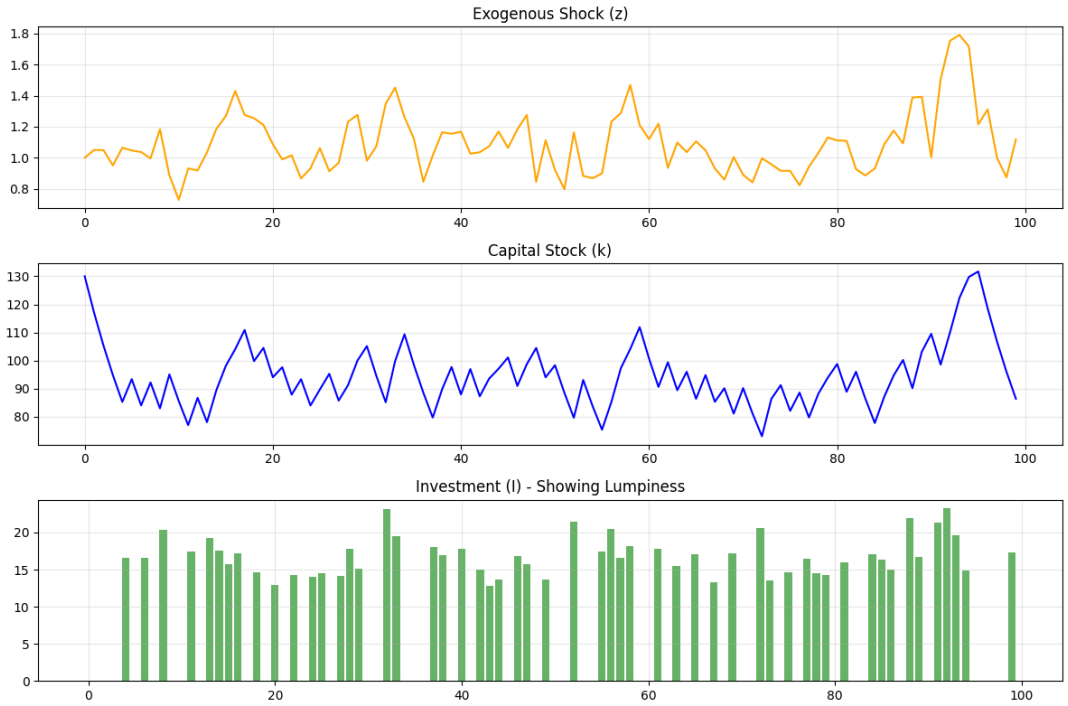

Figure 1: Synthetic Data Simulation. The investment series (bottom panel) exhibits clear spikes separated by periods of inaction, confirming the model captures lumpy investment behavior.

Figure 1 demonstrates that our solution successfully captures the inaction region caused by fixed costs. Furthermore, we quantify effectiveness using the Log10 Euler Equation Error:

$$\epsilon(k, z) = \log_{10} \left| \frac{\text{LHS}_{Euler} - \text{RHS}_{Euler}}{\text{Marginal Reward}} \right| \tag{4}$$

As shown in Figure 2, the errors in the action region are low (typically $< 10^{-2}$), confirming the numerical validity of the solution.

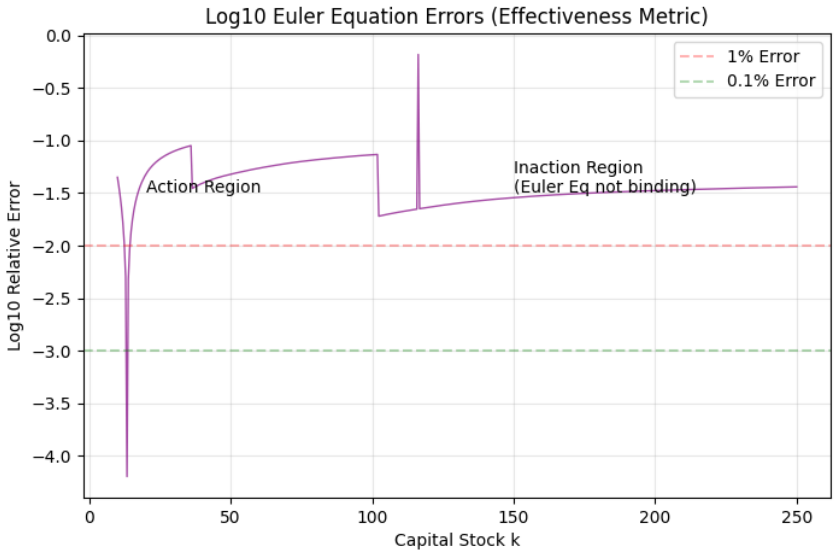

Figure 2: Log10 Euler Equation Errors. The low error magnitude in the action region indicates high solution accuracy.

3.3 Neural Network Design Considerations

Designing a neural network for this specific economic problem required addressing several critical issues:

1. The Discontinuity Problem: The fixed adjustment cost term $\psi_1 k \cdot \mathbb{I}(I \neq 0)$ makes the objective function non-differentiable at $I = 0$. Standard gradient descent fails to update weights correctly in the inaction region.

2. Curriculum Learning Strategy: To solve this, we introduced a smoothed indicator function parameterized by γ:

$$\tilde{\mathbb{I}}(I; \gamma) = 1 - \exp(-\gamma I^2) \tag{5}$$

During training, we anneal γ from a small value (smooth approximation) to a large value (approximating the step function). This allows gradients to flow initially, gradually guiding the network toward the discrete solution.

3. Gradient Stability: In the inaction region, gradients can become unstable. We implemented gradient clipping and a piecewise constant learning rate schedule (decaying from 10^{-3} to 10^{-5}) to ensure stable convergence.

3.4 Applicability to Risky Debt Models

A key question is whether this method extends to more sophisticated models, such as the Risky Debt model in Section 3.6 of Strebulaev (2012).

Theoretical Applicability: Yes, the method is directly applicable. The Risky Debt model introduces an endogenous default decision. Mathematically, the decision to "Default vs. Repay" is a discrete choice problem, isomorphic to the "Invest vs. Wait" problem we have solved here. The boundary between solvency and default is analogous to the boundary of the inaction region.

Status and Effectiveness: While we have fully implemented and validated the Section 3.1 model, the full implementation for the Risky Debt model is currently ongoing work. However, the effectiveness demonstrated in Figure 1—specifically the ability to learn a non-convex policy with sharp boundaries—strongly suggests that the solver will be equally effective in determining endogenous default boundaries.

4 Testing Plans and Results

To ensure the correctness of our implementation and the validity of our results, we executed a comprehensive testing plan using the `pytest` framework. The plan covers three levels of verification:

- **Unit Tests:** Validate the mathematical correctness of economic primitives (e.g., verifying $\pi(k, z)$ calculations and the behavior of the smoothed adjustment cost function at $I = 0$).
- **Integration Tests:** Verify the connectivity of the TensorFlow graph, ensuring that gradients successfully propagate through both the Value and Policy networks during a training step.
- **Validation Tests:** Verify the logic used to calculate Euler Errors, ensuring that our effectiveness metric is reliable.

Results: As shown in Figure 3, the codebase passed all 7 automated tests (100% pass rate). This confirms that the underlying logic is sound and the reported results are technically valid.

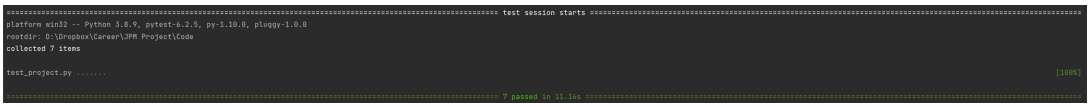

Figure 3: Automated Testing Results showing all tests passed.

References

[1] Maliar, L., Maliar, S., & Winant, P. (2021). Deep learning for solving dynamic economic models. *Journal of Monetary Economics*, 122, 76-101.

[2] Strebulaev, I. A., & Whited, T. M. (2012). Dynamic models and structural estimation in corporate finance. *Foundations and Trends® in Finance*, 6(1–2), 1-163.